Math

UIUCDCS-R-79-955

UILU-ENG 79 1701

Inference Rules for Unsatisfiability

by

David A. Plaisted

January 1979

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN · URBANA, ILLINOIS

UIUCDCS-R-79-955

Inference Rules for Unsatisfiability

by

David A. Plaisted

Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801

January 1979

This research was supported in part by the National Science Foundation under MCS 77-22830.

Abstract

There are some relationships between unsatisfiability of sets of clauses and properties of polynomials in several variables. These polynomials can be used to obtain a polynomial time solution to a certain problem involving sets of clauses. Using these polynomials, one can establish a correspondence between unsatisfiable sets of clauses and a convex region of Euclidean space. Also, some inference rules based on these polynomials may provide shorter proofs of inconsistency than are possible using other known inference rules.

Introduction

There are interesting relationships between the satisfiability problem and problems involving polynomials in several variables. The properties of such polynomials yield inference rules which may provide shorter proofs of inconsistency than resolution or other known inference rules can provide. Of course, if all inconsistent sets of clauses have short proofs, then NP = CoNP. Another possibility is that short proofs exist relative to a slowly growing, but infinite, set of axioms. We explore these possibilities. It turns out that polynomials associated with inconsistent sets of clauses over n variables correspond to a region of Euclidean space which is convex and is the intersection of 2^n half-spaces. We present polynomial time algorithms for several problems involving these polynomials, and present problems for which no polynomial time solution is known. This work contrasts with earlier work of the author [4] in which the satisfiability problem is related to sparse polynomials in one variable.

Polynomials in Many Variables

Definition: With a vector $\bar{x}$ in $\{0, 1\}^n$ we associate an interpretation $I(\bar{x})$ of the variables $x_1, x_2, \ldots, x_n$ in the usual way. That is, x_i is true in $I(\bar{x})$ if x_i (the i^{th} component of $\bar{x}$) is 1; x_i is false in $I(\bar{x})$ if x_i is 0.

Definition: Suppose S is a set of clauses over the variables $x_1, x_2, \ldots, x_n$ and f is a function assigning an integer weight to each clause in S. Then Poly(S, f) is defined to be the polynomial p over the

variables $x_1, x_2, \ldots, x_n$ having the following properties:

1. For all $\bar{x} \in \{0, 1\}^n$, $p(\bar{x}) = f(C_1) + f(C_2) + \ldots + f(C_k)$ where $\{C_1, C_2, \ldots, C_k\}$ is the set of clauses of S that are false in $I(\bar{x})$. We assume that the C_i are all distinct. Thus $p(\bar{x})$ is the weighted sum of the clauses of S that are false in the interpretation $I(\bar{x})$.
2. The polynomial p is a sum of terms of the form $x_{i_1} x_{i_2} \ldots x_{i_m}$ where $i_1, i_2, \ldots, i_m$ are all distinct. Thus no variable occu in p to a power higher than the first power.

It is not difficult to show using properties of polynomials of several variables that Poly(S, f) is uniquely defined, given S and f. Therefore $p(\bar{x}) = 0$ for all $\bar{x} \in \{0, 1\}^n$ iff all coefficients of p are zero. Also, if S is a set of 3-literal clauses, then Poly(S, f) can be computed from S and f in a number of arithmetic operations that is linear in the size of S.

<u>Examples</u>

$$\mathrm{Poly}(\{x_1 \vee x_2 \vee x_3\}, 1) = (1 - x_1)*(1 - x_2)*(1 - x_3)$$
$$= 1 - x_1 - x_2 - x_3 + x_1x_2 + x_2x_3 + x_1x_3 - x_1x_2x_3$$

$$\mathrm{Poly}(\{\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3\}, 1) = x_1x_2x_3$$

$$\mathrm{Poly}(S, f) = \Sigma_{C \in S} f(C)*\mathrm{Poly}(\{C\}, 1)$$

(We denote the constant function $f(\bar{x}) \equiv c$ by c; thus 1 denotes the constant function $f(\bar{x}) \equiv 1$.)

This construction gives an efficient algorithm for the following problem, first treated in [2]:

Problem P1: Given sets S1 and S2 of 3-literal clauses over $x_1, \ldots, x_n$, to decide whether for all interpretations I of $x_1, \ldots, x_n$, the number of clauses of S1 that are false in I equals the number of clauses of S2 that are false in I.

We solve this problem by computing Poly(S1, 1) and Poly(S2, 1). The sets S1 and S2 satisfy the above condition iff Poly(S1, 1) = Poly(S2, 1). This test only requires a number of arithmetic operations and comparisons that is linear in the size of S1 and S2. We still do not know whether problem P1 can be solved in polynomial time if the number of literals per clause is unbounded.

We can also get an efficient, trivial algorithm for the following problem, using these polynomials:

Problem P2: Given sets S1 and S2 of arbitrarily large negative clauses over $\{x_1, \ldots, x_n\}$, to decide whether for all interpretations I of $x_1, x_2, \ldots, x_n$, the number of clauses of S1 that are false in I equals the number of clauses of S2 that are false in I. (A clause is negative if all literals in the clause are negative. Thus $\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3$ is a negative clause.) This problem was posed in [2], in a different form.

Note that if S1 and S2 consist entirely of negative clauses, then Poly(S1, 1) and Poly(S2, 1) can be obtained very easily, and the above condition is true iff Poly(S1, 1) = Poly(S2, 1). However, it turns out that in this case Poly(S1, 1) = Poly(S2, 1) iff S1 = S2. Thus the condition is true iff S1 = S2. This problem is closely related to an NP-complete problem mentioned in [3]. If the number of positive literals per clause is bounded, we can compute Poly(S1, 1) and Poly(S2, 1) in linear time and so obtain a fast algorithm for this generalized problem.

In fact, we can still get a linear algorithm if the number of x_i such that x_i and $\bar{x}_i$ both appear in S1 $\cup$ S2 is bounded. This is because changing the sign of a propositional variable does not affect the property we are testing for. In particular, we can still solve Problem P2 efficiently if all the clauses are positive (that is, have only positive literals).

Theorem: The following problem is NP-complete:

Problem P3: Given a polynomial $p(x_1, x_2, \ldots, x_n)$ with integer coefficients, to determine whether there exists $\bar{x} \in \{0, 1\}^n$ such that $p(\bar{x}) = 0$.

Proof: This problem is clearly in NP. Also, a set S of 3-literal clauses over the variables $x_1, x_2, \ldots, x_n$ is consistent iff $\exists \bar{x} \in \{0, 1\}^n$ such that $\text{Poly}(S, 1)(\bar{x}) = 0$. Furthermore, Poly(S, 1) can be computed from S in polynomial time.

This result is not very profound, but polynomials in several variables have a convenient mathematical structure which helps to give us insight into the nature of the satisfiability problem.

Theorem: Suppose S1 and S2 are sets of 3-literal clauses over $x_1, x_2, \ldots, x_n$ and f_1 and f_2 are weighting functions for S1 and S2, respectively. Suppose $f_1(C) > 0$ for all $C \in S1$ and $f_2(C) > 0$ for all $C \in S2$. Suppose $\text{Poly}(S1, f_1) = \text{Poly}(S2, f_2)$. Then S1 is inconsistent iff S2 is.

This result suggests inference rules for unsatisfiability. Namely, if we know S1 as in the theorem is inconsistent, so is S2. However, there does not appear to be any relationship between such sets S1 and S2 in terms of proofs of inconsistency. Hence we might hope to

obtain short proofs of inconsistency using inference rules based on Poly(S, f) for a set S of 3-literal clauses.

For example, the following sets S1, S2 and S3 of clauses satisfy Poly(S1, 1) = Poly(S2, 1) = Poly(S3, 1) = 1:

S1: $x_1 \vee x_2 \vee x_3$ $\quad$ $\bar{x}_1 \vee x_2 \vee x_3$

$x_1 \vee x_2 \vee \bar{x}_3$ $\quad$ $\bar{x}_1 \vee x_2 \vee \bar{x}_3$

$x_1 \vee \bar{x}_2 \vee x_3$ $\quad$ $\bar{x}_1 \vee \bar{x}_2 \vee x_3$

$x_1 \vee \bar{x}_2 \vee \bar{x}_3$ $\quad$ $\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3$

S2: $x_1 \vee x_2 \vee x_3$ $\quad$ $x_1 \vee \bar{x}_2$

$\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3$ $\quad$ $x_2 \vee \bar{x}_3$

$x_3 \vee \bar{x}_1$

S3: x_1

$\bar{x}_1 \vee x_2$

$\bar{x}_1 \vee \bar{x}_2 \vee x_3$

$\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3$

Thus if we know one of these sets to be inconsistent, we can easily show the others to be inconsistent since their polynomials are all identical.

Linearity Properties of the Coefficients

Notice that the coefficients of Poly(S1, f) for fixed S1 are linear combinations of the values f(C) for C ε S1. Thus we can get a polynomial time solution to the following problem.

Problem P4: Given sets S1 and S2 of 3-literal clauses over $x_1, x_2, \ldots, x_n$, to find weighting functions f_1 and f_2 such that $\mathrm{Poly}(S1, f_1) = \mathrm{Poly}(S2, f_2)$, if such f_1 and f_2 exist.

This problem can be solved in polynomial time since $\mathrm{Poly}(S1, f_1) = \mathrm{Poly}(S2, f_2)$ iff all coefficients of $\mathrm{Poly}(S1, f_1) - \mathrm{Poly}(S2, f_2)$ are zero. Also, each coefficient of $\mathrm{Poly}(S1, f_1) - \mathrm{Poly}(S2, f_2)$ is a linear combination of the values $f_1(C)$ for C in S1 and $f_2(C)$ for C in S2. Hence we have a set of homogeneous linear equations. We can find a solution (if it exists) by Gaussian elimination. The solution will be rational, so by multiplying through by suitable integers we can obtain an integer solution.

By similar methods, we get a polynomial time algorithm for the following problem:

Problem P5: Given sets S1 and S2 of 3-literal clauses over $x_1, \ldots, x_n$, and given a weighting function f_1 for S1, to find an integer $k \neq 0$ and a weighting function f_2 for S2 such that $k*\mathrm{Poly}(S1, f_1) = \mathrm{Poly}(S2, f_2)$, if such k and f_2 exist.

The significance of this result is that if for all $\bar{x} \in \{0, 1\}^n$, $\mathrm{Poly}(S1, f_1)(\bar{x}) \neq 0$, and if such k and f_2 exist, then S2 is inconsistent. Further, if S1 is inconsistent and $f_1(C) > 0$ for all C in S1, then $\mathrm{Poly}(S1, f_1)(\bar{x}) \neq 0$ for all $\bar{x} \in \{0, 1\}^n$.

The following problem is related but is harder.

Problem P6: Given sets S1 and S2 of 3-literal clauses over $x_1, x_2, \ldots, x_n$, and given weighting function f_1 for S1, to find a weighting function f_2 for S2 such that $\mathrm{Poly}(S1, f_1) = \mathrm{Poly}(S2, f_2)$, if such f_2 exists. By previous remarks, this is equivalent to solving a

Correction to Page 7 of "Inference Rules for Unsatisfiability" by David A. Plaisted

Problem P4: Given sets S1 and S2 of 3-literal clauses over $x_1, \ldots, x_n$, and given a weighting function f_1 for S1, to find an integer $k \neq 0$ and a weighting function f_2 for S2 such that $k^*\mathrm{Poly}(S1, f_1) = \mathrm{Poly}(S2, f_2)$, if such k and f_2 exist.

We can solve this problem in polynomial time since each coefficient of $\mathrm{Poly}(S2, f_2)$ is a linear combination of the values $f_2(C)$ for $C \in S2$. By Gaussian elimination, we can obtain rational values for the quantities f(C) so that $\mathrm{Poly}(S2, f) = \mathrm{Poly}(S1, f_1)$, if such values exist. Let $k \neq 0$ be an integer so that kf(C) is an integer for all $C \in S2$, and let $f_2(C)$ be kf(C) for all $C \in S2$. Then $\mathrm{Poly}(S2, f_2) = k^*\mathrm{Poly}(S1, f_1)$.

The significance of this result is that if for all $\bar{x} \in \{0, 1\}^n$, $\mathrm{Poly}(S1, f_1)(\bar{x}) \neq 0$, and if such k and f_2 exist, then S2 is inconsistent. Further, if S1 is inconsistent and $f_1(C) > 0$ for all C in S1, then $\mathrm{Poly}(S1, f_1)(\bar{x}) \neq 0$ for all $\bar{x} \in \{0, 1\}^n$.

Consider the following problem.

Problem P5: Given sets S1 and S2 of 3-literal clauses over $x_1, x_2, \ldots, x_n$, to find weighting functions f_1 and f_2 such that $f_1(C) \geq 1$ for all $C \in S1$ and such that $\mathrm{Poly}(S1, f_1) = \mathrm{Poly}(S2, f_2)$, if such f_1 and f_2 exist.

The significance of this problem is that if S1 is inconsistent and if f_1 and f_2 exist, then S2 is inconsistent also. Although we do not have a polynomial time solution to problem P5, we have the following easy result:

<u>Theorem</u>: Problem P5 can be polynomially reduced to the following problem: Given an integer matrix A and an integer ℓ, to determine whether there exists a vector $\bar{z}$ such that $A\bar{z} = 0$ and such that $z_i \geq 1$ for $i = 1, 2, \ldots, \ell$.

The following problem appears to be easier.

Problem P6: ...

Correction to Page 8 line -5: $\bar{x}$ should be $\bar{z}$ two places.

set of non-homogeneous linear diophantine equations. Techniques for solving such systems can be found in [1].

Definition: A rational weighting function f for a set S of clauses is a function from elements of S to rational numbers. That is, the weight of each clause may be a rational number. The usual kind of weighting function will be called an integer weighting function when necessary. Weighting functions will be assumed to be integer weighting functions unless otherwise specified.

Consider the following problem:

Problem P7: Given sets S1 and S2 of 3-literal clauses and an integer weighting function f_1 for S1, to determine if there exists a rational weighting function f_2 for S2 such that every coefficient of Poly(S2, f_2) - Poly(S1, f_1) is nonnegative. Note that if such an f_2 exists, and if Poly(S1, f_1)($\bar{x}$) > 0 for all $\bar{x} \in \{0, 1\}^n$, then Poly(S2, f_2)($\bar{x}$) > 0 for all $\bar{x} \in \{0, 1\}^n$ also and so S2 is inconsistent.

We can easily get the following result.

Theorem: Problem P7 can be polynomially reduced to the following problem: Given an integer matrix A and an integer vector $\bar{b}$, to determine if there exists a vector $\bar{x}$ such that $A\bar{x} \geq \bar{b}$. Here inequality is applied componentwise.

Isomorphism

Definition: Suppose S1 and S2 are sets of 3-literal clauses over $x_1, x_2, \ldots, x_n$. We say S1 $\sim$ S2 if S2 can be obtained from S1 by permuting variables and by changing signs of variables.

It is clear that if S1 $\sim$ S2, then S1 is inconsistent iff S2 is. Also, it is not hard to show that determining whether S1 $\sim$ S2 is polynomially equivalent to graph isomorphism. Similarly, given polynomials p_1 and p_2 over $x_1, \ldots, x_n$, determining whether p_1 can be obtained from p_2 by permuting variables, is polynomially equivalent to graph isomorphism. We do not know whether this is still true if we also allow replacements of the form $x_j \leftarrow 1-x_j$.

<u>Definition</u>: Suppose p_1 and p_2 are polynomials in the variables $x_1, x_2, \ldots, x_n$. We say $p_1 \sim p_2$ if p_2 can be obtained from p_1 by permuting variables and by replacements of the form $x_j \leftarrow 1-x_j$. Note that this is an equivalence relation.

Denseness of Non-zero Values

The following results give us more insight into the behavior of the functions Poly(S, f). In particular, the values of Poly(S, f)($\bar{x}$) on all $\bar{x}$ in $\{0, 1\}^n$ are determined by the values at a small set of such $\bar{x}$, as we will show. Let R be the set of real numbers.

<u>Definition</u>: Suppose $\bar{x}, \bar{y} \in \{0, 1\}^n$. We say $\bar{x} \leq \bar{y}$ if for $i = 1, 2, \ldots, n$, $x_i \leq y_i$.

<u>Definition</u>: If $\bar{x}$ is an n-tuple of real numbers, then $||\bar{x}||$ is $\sum_{i=1}^{n} |x_i|$.

<u>Definition</u>: Suppose q is a function from R^n into R. Then $\Delta_i q$ is the function defined by $(\Delta_i q)(x_1, \ldots, x_n) = q(x_1, x_2, \ldots, x_{i-1}, x_i + 1, x_{i+1}, \ldots, x_n) - q(x_1, \ldots, x_n)$. Similarly, $\Delta_j \Delta_i q$ is defined.

<u>Definition</u>: Suppose $\bar{x} \in \{0, 1\}^n$. Suppose q is a function from R^n into R. Then $\Delta_{\bar{x}} q$ is $\Delta_{i_1} \Delta_{i_2} \ldots \Delta_{i_k} q$ where $i_1 < i_2 < \ldots < i_k$ and $\{i_1, i_2, \ldots, i_k\} = \{j: x_j = 1\}$.

<u>Theorem</u>: Suppose S is a set of 3-literal clauses over $x_1, x_2, \ldots, x_n$ and f is a weighting function for S. Suppose $\bar{x} \in \{0, 1\}^n$ and $||x|| \geq 4$. Then $\Delta_{\bar{x}}\text{Poly}(S, f) \equiv 0$.

<u>Proof</u>: The degree of $\Delta_i p$ is at most one less than the degree of p, unless p is a constant. Also, the degree of Poly(S, f) is 3 or less. Hence if $||\bar{x}|| = 3$, the degree of $\Delta_{\bar{x}}\text{Poly}(S, f)$ is 0 and $\Delta_{\bar{x}}\text{Poly}(S, f)$ is a constant. Therefore if $||\bar{x}|| = 4$, $\Delta_{\bar{x}}\text{Poly}(S, f) \equiv 0$.

<u>Theorem</u>: Suppose S is a set of 3-literal clauses over $x_1, x_2, \ldots, x_n$ and f is a weighting function for S. Suppose Poly(S, f) is not identically zero. Then there exists $\bar{x} \in \{0, 1\}^n$ such that $||\bar{x}|| \leq 3$ and such that $\text{Poly}(S, f)(\bar{x}) \neq 0$.

<u>Proof</u>: Let $\bar{y}$ be a minimal vector in $\{0, 1\}^n$ such that $\text{Poly}(S, f)(\bar{y}) \neq 0$. Then $\Delta_{\bar{y}}\text{Poly}(S, f)(0, 0, \ldots, 0) \neq 0$. Hence $||\bar{y}|| \leq 3$.

<u>Corollary</u>: Poly(S, f) is completely determined by the $\binom{n}{3} + \binom{n}{2} + n + 1$ values $\text{Poly}(S, f)(\bar{x})$ for $||\bar{x}|| \leq 3$.

It follows that $\text{Poly}(S, f) \equiv 0$ if $\text{Poly}(S, f)(\bar{x}) = 0$ for all $\bar{x} \in \{0, 1\}^n$ with $||\bar{x}|| \leq 3$. In fact, if Poly(S, f) is not identically zero, then for <u>all</u> $\bar{y} \in \{0, 1\}^n$, there exists $\bar{x} \in \{0, 1\}^n$ such that $||\bar{x} - \bar{y}|| \leq 3$ and such that $\text{Poly}(S, f)(\bar{x}) \neq 0$. Thus interpretations giving non-zero values are "dense". It does not follow, however, that $\text{Poly}(S, f)(\bar{x}) \geq 0$ for all $\bar{x} \in \{0, 1\}^n$ iff $\text{Poly}(S, f)(\bar{x}) \geq 0$ for all $\bar{x} \in \{0, 1\}^n$ with $||\bar{x}|| \leq 3$. For example, let S be the set of $2\binom{n}{3}$ clauses over $x_1, x_2, \ldots, x_n$ in which $x_i \vee x_j \vee x_k$ and $\bar{x}_i \vee \bar{x}_j \vee \bar{x}_k$ occur in S for all i, j, k with $i < j < k$. Define f by $f(C) = -1$ on clauses C of the form $\bar{x}_i \vee \bar{x}_j \vee \bar{x}_k$ and $f(C) = 1$ on clauses C of the form $x_i \vee x_j \vee x_k$. Suppose $n \geq 7$. Then $\text{Poly}(S, f)(\bar{x}) > 0$ for all

$\bar{x} \in \{0, 1\}^n$ with $||\bar{x}|| \leq 3$, but Poly(S, f)(1, 1, ..., 1) is $-\binom{n}{3}$.

The Unsatisfiable Region of Euclidean Space

Definition: Let M be the set of polynomials p with real coefficients over the variables $x_1, \ldots, x_n$ such that p can be expressed as a sum of terms of one of the following forms, for $i < j$ and $j < k$:

$a_{ijk}x_ix_jx_k$

$b_{ij}x_ix_j$

c_ix_i

d

Note that such a polynomial is specified by $\binom{n}{3} + \binom{n}{2} + n + 1$ real coefficients. We thus identify polynomials in M with points in N-dimensional Euclidean space, where $N = \binom{n}{3} + \binom{n}{2} + n + 1$. Usually we are interested in the set of integer coefficient polynomials of M.

We would like to know which region of R^N corresponds to polynomials $p \in M$ such that $p(\bar{x}) > 0$ for all $\bar{x} \in \{0, 1\}^n$. Such polynomials represent inconsistent sets of clauses. Therefore we have the following definitions.

Definition: Let UNSATP be $\{p \in M: p(\bar{x}) > 0$ for all $\bar{x} \in \{0, 1\}^n\}$. We also use UNSATP to refer to the corresponding subset of R^N.

Theorem: UNSATP is a convex region of R^N. In fact, UNSATP is the intersection of 2^n half-spaces in R^N. Also, if $\bar{z} \in$ UNSATP then $a\bar{z} \in$ UNSATP for all $a > 0$.

Proof: Let $\bar{w}$ be an element of $\{0, 1\}^n$. Suppose p is a polynomial in UNSATP. Then $p(\bar{w})$ is a sum of coefficients of p. Hence $\{p \in M: p(\bar{w}) > 0\}$ is a half-space of R^N. Therefore UNSATP is the intersection of 2^n half-spaces in R^N. Since each half-space is convex,

so is UNSATP. Also, if $p(\bar{w}) > 0$ then $ap(\bar{w}) > 0$ for all $a > 0$. Hence $p \in$ UNSATP implies $ap \in$ UNSATP for all $a > 0$.

Using these results, we might be able to verify that a point is in UNSATP by exhibiting a perpendicular to a suitable hyperplane bounding UNSATP.

Inference Rules

We now show how the polynomials associated with sets of clauses can be used to obtain more inference rules for unsatisfiability. That is, we obtain inference rules that can be used to show that a set of clauses is unsatisfiable. It is conceivable that the use of these rules, together with other inference rules such as resolution, will make possible much shorter proofs than are possible without using these rules. Therefore, this work is closely related to the NP vs. CoNP question.

We use $GE(p_1, p_2)$ to abbreviate $(\forall \bar{x} \in \{0, 1\}^n)\ p_1(\bar{x}) \geq p_2(\bar{x})$. Also, the polynomial whose value is the constant k is written k. Thus $GE(p, 1)$ means $(\forall \bar{x} \in \{0, 1\}^n) p(\bar{x}) \geq 1$. Further, if f is a weighting function for a set S of clauses, then $f \geq k$ abbreviates $(\forall C \in S) f(C) \geq k$. Similarly, $f > k$ abbreviates $(\forall C \in S) f(C) > k$. Note that a set S of clauses is inconsistent iff $(\exists f) GE(Poly(S, f), 1)$. We introduce inference rules involving expressions of the form $GE(p, q)$.

List of Inference Rules

Group 1

1. $Poly(S, f_1 + f_2) = Poly(S, f_1) + Poly(S, f_2)$
2. $Poly(S, kf) = k*Poly(S, f)$
3. $S1 \cap S2 = \emptyset \supset Poly(S1 \cup S2, f) = Poly(S1, f) + Poly(S2, f)$
4. (S is inconsistent) iff $(\exists f)\ GE(Poly(S, f), 1)$
5. (S is inconsistent) iff $(\exists f)\ f \geq 0 \wedge GE(Poly(S, f), 1)$

6. $f > 0 \supset [(S \text{ is inconsistent}) \text{ iff } GE(Poly(S, f), 1)]$
7. $S1 \subset S2 \wedge f > 0 \supset GE(Poly(S2, f), Poly(S1, f))$
8. $f_1 > 0 \wedge f_2 > 0 \supset [GE(Poly(S, f_1), 1) \equiv GE(Poly(S, f_2), 1)]$
9. $f_1 > 0 \wedge f_2 > 0 \wedge Poly(S1, f_1) = Poly(S2, f_2) \supset S1 \equiv S2$
10. $S1 \equiv S2 \supset (S \cup S1 \text{ is inconsistent}) \text{ iff } (S \cup S2 \text{ is inconsistent})$

Group 2

1. $GE(p, p)$
2. $GE(p, q) \wedge GE(q, r) \supset GE(p, r)$
3. $GE(p_1, q_1) \wedge GE(p_2, q_2) \supset GE(p_1 + q_1, p_2 + q_2)$
4. $GE(p, q)$ iff $GE(-q, -p)$
5. $GE(q_1, 0) \wedge GE(q_2, 0) \wedge GE(p_1, q_1) \wedge GE(p_2, q_2) \supset GE(p_1 * p_2, q_1 * q_2)$
6. $k1 \geq 0 \wedge k_2 \geq 0 \wedge GE(p_1, k_1) \wedge GE(p_2, k_2) \supset GE(p_1 * p_2, k_1 * k_2)$
7. $k > 0 \supset [GE(p, q) \equiv GE(kp, kq)]$
8. $GE(q, 1) \supset [GE(p_1, p_2) \equiv GE(p_1 * q, p_2 * q)]$
9. $GE(x_i, 0)$ for $1 \leq i \leq n$ and $GE(1 - x_i, 0)$ for $1 \leq i \leq n$
10. $GE(x_i, x_i^k)$ for $1 \leq i \leq n$, $k > 0$
11. $GE(x_i^k, x_i)$ for $1 \leq i \leq n$, $k > 0$

Group 3

1. $S1 \sim S2 \supset S2 \sim S1$
2. $p_1 \sim p_2 \supset p_2 \sim p_1$
3. $S1 \sim S2 \supset [(S1 \text{ is inconsistent}) \text{ iff } (S2 \text{ is inconsistent})]$
4. $p_1 \sim p_2 \supset k * p_1 \sim k * p_2$
5. $GE(p_1, q) \wedge p_1 \sim p_2 \supset GE(p_2, q)$
6. $S1 \sim S2 \wedge f_1 > 0 \supset [(\exists f_2) f_2 > 0 \wedge Poly(S1, f_1) \sim Poly(S2, f_2)]$

We now illustrate ways in which these rules can be used. Suppose S1 is inconsistent and $S1 \subset S2$. Then by 1.6, GE(Poly(S1, 1), 1). Also, by 1.7, GE(Poly(S2, 1), Poly(S1, 1)). Hence by 2.2, GE(Poly(S2, 1), 1). Hence by 1.6, S2 is inconsistent. Thus we only need to worry about minimal inconsistent sets of clauses. These can be reduced in number by 3.3. In addition, from 2.5, 2.7, and 2.9 it follows that GE(p, 0) is true if all coefficients of p are nonnegative. Also, it follows from 2.5, 2.7, and 2.9 that GE(Poly(S, f), 0) for all S if $f \geq 0$. Suppose S1 is a minimal inconsistent set of clauses, and for some weighting function f_1, $Poly(S1, f_1) = Poly(S2, f_2) + p$ where $f_2 > 0$ and GE(p, 0) is known.

Suppose S2 is knownto be inconsistent. Then it follows by 1.6 that GE(Poly(S2, f_2), 1) and by 2.3 that GE(Poly(S1, f_1), 1) and by 1.4 that S1 is inconsistent. Hence we may be able to exhibit short proofs of inconsistency of minimal inconsistent sets of clauses by methods other than isomorphism. Also, it could be that distinct minimal inconsistent sets S1 and S2 of clauses will have the same polynomials Poly(S1, f_1) = Poly(S2, f_2), and in this way we may get short proofs of inconsistency. Finally, the rules 2.10 and 2.11 can be used to eliminate powers of x_i higher than the first power after applying 2.5 or 2.6. The rules 2.5 or 2.6 will usually result in polynomials of degree higher than 3, even after such reduction in exponents has been done.

The following limited results concern minimal inconsistent sets of clauses.

Theorem: Suppose S1 and S2 are minimal inconsistent sets of clauses over $x_1, x_2, \ldots, x_k$. That is, no proper subset of S1 or S2 is inconsistent. Suppose $f_1 > 0$ and $f_2 > 0$ and Poly(S1, f_1) = Poly(S2, f_2). Then min $\{f_1(C): C \in S1\}$ = min $\{f_2(C): C \in S2\}$.

Proof: Let C1 $\in$ S1 be a clause such that f_1(C1) is minimal among $\{f_1(C): C \in S1\}$. Let C2 $\in$ S2 be a clause such that f_2(C2) is minimal among $\{f_2(C): C \in S2\}$. Since S1 is minimal inconsistent, S1 - {C1} is consistent and so some interpretation makes all clauses in S1 - {C1} true. Thus there exists $\bar{x} \in \{0, 1\}^n$ such that Poly(S1, f_1)($\bar{x}$)= f_1(C1). Hence Poly(S2, f_2)($\bar{x}$) = f_1(C1) also. Since Poly(S2, f_2)($\bar{x}$) is a sum of weights of clauses in S2, $f_1(C1) \geq f_2(C2)$. Similarly, $f_2(C2) \geq f_1(C1)$.

<u>Theorem</u>: Suppose S is a minimal inconsistent set of clauses over $x_1, x_2, \ldots, x_n$. Suppose f is a weighting function. Then GE(Poly(S, f), 1) is true iff $f > 0$.

<u>Proof</u>: If $f > 0$, GE(Poly(S, f), 1) follows because S is inconsistent. If for some $C \varepsilon S$, $f(C) \leq 0$ then GE(Poly(S, f), 1) is false, as follows: Since S is minimal inconsistent, there is an interpretation in which C is false and all other clauses of S are true. Hence there exists $\bar{x} \varepsilon \{0, 1\}^n$ such that $\text{Poly}(S, f)(\bar{x}) = f(C)$. Since $f(C) \leq 0$, we cannot have GE(Poly(S, f), 1).

There is still another technique that may be applied to show inconsistency. Let f be a weighting function for S such that for no nonempty subset {C1, C2, ..., Ck} of k distinct elements of S does $f(C1) + f(C2) + \ldots + f(Ck) = 0$. Such weighting functions can be obtained from instances of the knapsack problem or the partition problem that are known not to have a solution. And such instances can be obtained by polynomial time reductions from known inconsistent sets of clauses! In any event, if f is such a weighting function, and S is inconsistent, then $(\forall \bar{x} \varepsilon \{0, 1\}^n)\ \text{Poly}(S, f)(\bar{x}) \neq 0$. Hence if S1 is another set of clauses and f_1 is a weighting function for S1, and if $\text{Poly}(S1, f_1) = \text{Poly}(S, f)$, then S1 is inconsistent also. Such a function f need not satisfy $f \geq 0$, and so we get a more general method than that of rules 1.4, 1.5, and 1.6.

Finally, it would be interesting to know if there is a "small" set A_n of axioms from which the inconsistency of all inconsistent sets of 3-literal clauses over $x_1, \ldots, x_n$ can be shown by short proofs.

These axioms would be of the form GE(Poly(S, f), 1) for various S and f or of the form GE(p, 0) for various p. If so, unsatisfiability could be decided in nondeterministic polynomial time relative to a "slowly utilized" oracle [2]. Along this line, how many distinct polynomials p are there in the set IP = {Poly(S, 1): S is a minimal inconsistent set of clauses over $x_1, \ldots, x_n$}? How many equivalence classes are there in this set under the relationship $p_1 \sim p_2$?

Not all of these equivalence classes are really necessary. Suppose we eliminate from IP all equivalence classes of polynomials p satisfying the following condition:

> There exist S1, S2, f_1, f_2, q such that p = Poly(S1, 1) and S1, S2 are minimal inconsistent sets of clauses and Poly(S1, f_1) = Poly(S2, f_2) + q and $f_2 > 0$ and it is known that GE(q, 0) is true.

If this condition is true, then given that S2 is known to be inconsistent we can construct a short proof that S1 is inconsistent. Hence GE(Poly(S1, 1), 1) need not be kept as an axiom. The polynomial q may have nonnegative coefficients, or be of the form Poly(S, f) - 1 where S is known to be inconsistent and f > 0. Also, we can eliminate from IP all equivalence classes of polynomials Poly(S, 1) such that S has a short resolution proof of inconsistency. How many equivalence classes are then left in IP? If this number is small, we might hope to get short proofs of inconsistency relative to a small number of axioms.

Conclusions

Polynomials with several variables give insight into the structure of unsatisfiable sets of clauses. The polynomials associated with sets of clauses seem to have properties that do not have any relationship to difficulty of proving inconsistency of the sets of clauses.

It is possible, therefore, that these polynomials will provide methods of obtaining short proofs of inconsistency. It turns out that polynomials of unsatisfiable sets of clauses correspond to a region of Euclidean space which is the intersection of 2^n half-spaces, for sets of clauses over n variables. Some inference rules based on these polynomials can be used to show that a set of clauses is unsatisfiable. Several problems associated with these polynomials have polynomial time solutions.

References

[1] Niven, I. and H. Zuckerman. An Introduction to the Theory of Numbers. Wiley, New York, 1972, pp. 100-105.

[2] Plaisted, D. New NP-hard and NP-complete polynomial and integer divisibility problems. Proceedings of the 18th Annual Symposium on Foundations of Computer Science, 1977, pp. 241-253.

[3] Plaisted, D. Some polynomial and integer divisibility problems are NP-hard. SIAM J. Comput. 7, 1978, pp. 458-464.

[4] Plaisted, D. Sparse complex polynomials and polynomial reducibility. J. Comput. System Sci. 14, 1977, pp. 210-221.

BLIOGRAPHIC DATA EET	1. Report No. UIUCDCS-R-79-955	2.	3. Recipient's Accession No.
Title and Subtitle "Inference Rules for Unsatisfiability"			5. Report Date January 1979
			6.
Author(s) David A. Plaisted			8. Performing Organization Rept. No.
Performing Organization Name and Address Department of Computer Science, University of Illinois, Urbana, Illinois 61801			10. Project/Task/Work Unit No.
			11. Contract/Grant No. MCS 77-22830
Sponsoring Organization Name and Address National Science Foundation, Washington, D.C.			13. Type of Report & Period Covered
			14.

Supplementary Notes

Abstracts

There are some relationships between unsatisfiability of sets of clauses and properties of polynomials in several variables. These polynomials can be used to obtain a polynomial time solution to a certain problem involving sets of clauses. Using these polynomials, one can establish a correspondence between unsatisfiable sets of clauses and a convex region of Euclidean space. Also, some inference rules based on these polynomials may provide shorter proofs of inconsistency than are possible using other known inference rules.

Key Words and Document Analysis. 17a. Descriptors

Unsatisfiability, propositional calculus, polynomials in several variables, proof length, CoNP, Inference rules.

. Identifiers/Open-Ended Terms

. COSATI Field/Group

Availability Statement	19. Security Class (This Report) UNCLASSIFIED	21. No. of Pages
	20. Security Class (This Page) UNCLASSIFIED	22. Price